Network Working Group
Request for Comments #186

NIC 7130
DMCG

IDENTIFICATION

A Network Graphics Loader

James C. Michener

12 July 1971

MOTIVATION

The facility described herein will permit remote users on the ARPA network to obtain graphics output from programs they write for the Evans and Sutherland Line Drawing System Model 1 (LDS-1) located at the DMCG computer. Also, users at that computer can employ the facility to do graphics on their ARDS and IMLAC consoles.

INTRODUCTION

The Graphics Loader on the Project MAC Dynamic Modeling/Computer Graphics PDP-1Ø is for use with the E&S LDS-1 display. Display programs can be shipped to it and executed repeatedly. The output, which would normally be visible at the PDP-1Ø installation, is transmitted to the originating site in digital form.

Corrections and alterations to display programs can be transmitted so that the bulk of the program need be sent only once. Any data or parameters which vary may be sent whenever they change.

The originating site may request to have any part of its program or data transmitted back to it from the Graphics Loader.

With this feature it is possible to debug a display program which is incorrectly modifying itself.

In order to simplify the Graphics Loader, it is assumed that the display program should occupy a contiguous block of core starting at location 1000 octal (i.e., it has been assembled absolutely), that its first executable instruction is at the same place, and that, when one frame is complete, it jumps to location 777 octal.

The E&S LDS-1 has the capability of writing into memory the coordinates of endpoints of the line segments which would be visible to a user sitting at the LDS-1 display device. A register called the Write Address Register (WAR) is used to indicate an area of memory to contain these coordinates. Various submodes are available for output to memory, but for the submode of greatest interest, "Scaled Coordinates to Memory" mode, each "visible" line segment causes two words of coordinate data to be stored. The contents of the WAR are incremented for each word stored.

For each execution of the display program, the Graphics Loader sets the proper output mode (suppressing output to the cathode ray tube at the DGSD machine), initializes the WAR before execution and saves the final value of the WAR after execution. Thus it is easy for the Graphics Loader to transmit to the user only the "visible output" of the display program.

DESCRIPTION OF REQUESTS FROM THE REMOTE USER PROGRAM

Requests are in the form of 36 bit words. The first word of a request is interpreted as two 18 bit fields. The left half contains a number identifying which of six operations is being requested. The right half is either a mode or is ignored, depending on the requested operation. (If the left half is not a valid operation number, an error message is sent and the next word is considered to begin a new request.)

Depending upon the operation requested and upon the mode, one word of argument data may or may not be read. This word is (also) treated as two 18 bit halves. The interpretation of the halves depends on the operation. In the description of individual operations, the left half will be called A1; the right half will be called A2 (standing for Arguments 1 and 2).

Error checking of the arguments is performed next. If an error condition is present, error information is sent to the user program at the originating site and the Graphics Loader prepares itself for the next command. If no error condition is present, an acknowledging message is sent unless the Suppress Acknowledgement mode prevails. For certain requests, the operation is performed before the acknowledgement is transmitted.

For those operations involving a transfer of display program information (either to or from the Graphics Loader), this transfer is done next, after the error checking of arguments has been performed and after an acknowledge message has been sent.

This done, the Graphics Loader reads the next command.

SPECIFICATIONS

0. The valid operations are currently:

SETUP indicated by an operation number of 1

EXECUTE indicated by an operation number of 2

TRANSMIT indicated by an operation number of 3

UPDATE indicated by an operation number of 4

FLUSH connection indicated by an operation number of 5

MODESET indicated by an operation number of 6

An invalid operation number is an error condition (condition number Ø).

1. The SETUP request.

The mode field of the first word is ignored. SETUP requires an argument word. The arguments A1 and A2 are both treated like lengths, so both must be non-negative numbers. If they are not, error condition 2 is recognized.

1A. If A1 is strictly positive, then this request describes a whole new display program, and any previous display program from this user is to be forgotten. In this case, A1 is the total length of the display program, exclusive of the area to be addressed by the Write Address Register (WAR). A2 is the length of the area to be addressed by the WAR. As such, A2 must be at least twice the greatest possible number of visible line segments to be displayed. (If the LDS-1 programmer feels sure of himself, he may set up his own "WAR area" and set his own "Output To Memory" modes. He would not need to use the A2 parameter at all.)

An acknowledge message is sent (unless suppressed; see MODESET). Then the display program is read (which consists of (A1) words).

1B. If A1 is zero then this request is for a change in the length of the area to be addressed by the WAR. A2 contains the new length. A2 may be larger or smaller than the current length of the area. If no previous 1A type of SETUP request has been processed, error condition 1 is recognized. Otherwise an acknowledge message is sent (unless suppressed, see MODESET).

(This request would typically be used if an initial estimate on the number of words required were too low. Error condition 5, described under EXECUTE may indicate a low estimate.)

2. The EXECUTE request.

The EXECUTE request does not take a parameter word, but the mode field is used to specify the number of times that the "EXECUTE action" is to be performed. (This "action" is described in detail following this paragraph. Briefly, it is a single execution of the display program.) If the mode field is zero or negative, then one EXECUTE action is performed. Whenever an error is encountered during a multiple EXECUTE, the iteration is immediately stopped. This way, the status of the display program after the error is not destroyed, and no flood of error messages is ever sent, only a single one.

The EXECUTE action is as follows:

If no previous SETUP request has been processed, error condition number 1 will be recognized.

An attempt is made to seize the E&S LDS-1 display processor. (If a previous EXECUTE has succeeded in seizing it, then this will also succeed. If some other user of the DMCG machine has control of the display processor, this will fail and error condition 4 will be recognized.)

The display program is now executed. The environment at the beginning of execution of the display program is given in Appendix I.

If a previous execute failed in a way indicating a programming error in the display program (error conditions 6, 7 and 8), then an SETUP or an UPDATE request must be executed before another EXECUTE request will be processed. (If no SETUP or UPDATE is given before another EXECUTE, then error condition 3 is recognized.)

If the LDS-1 runs for two seconds without causing an interrupt or jumping to the "finish" location (the word before the origin of the display program) then it is assumed the program is running away. The LDS-1 is stopped, and error condition 6 is recognized. (A SETUP or an UPDATE is required before another EXECUTE is permitted.)

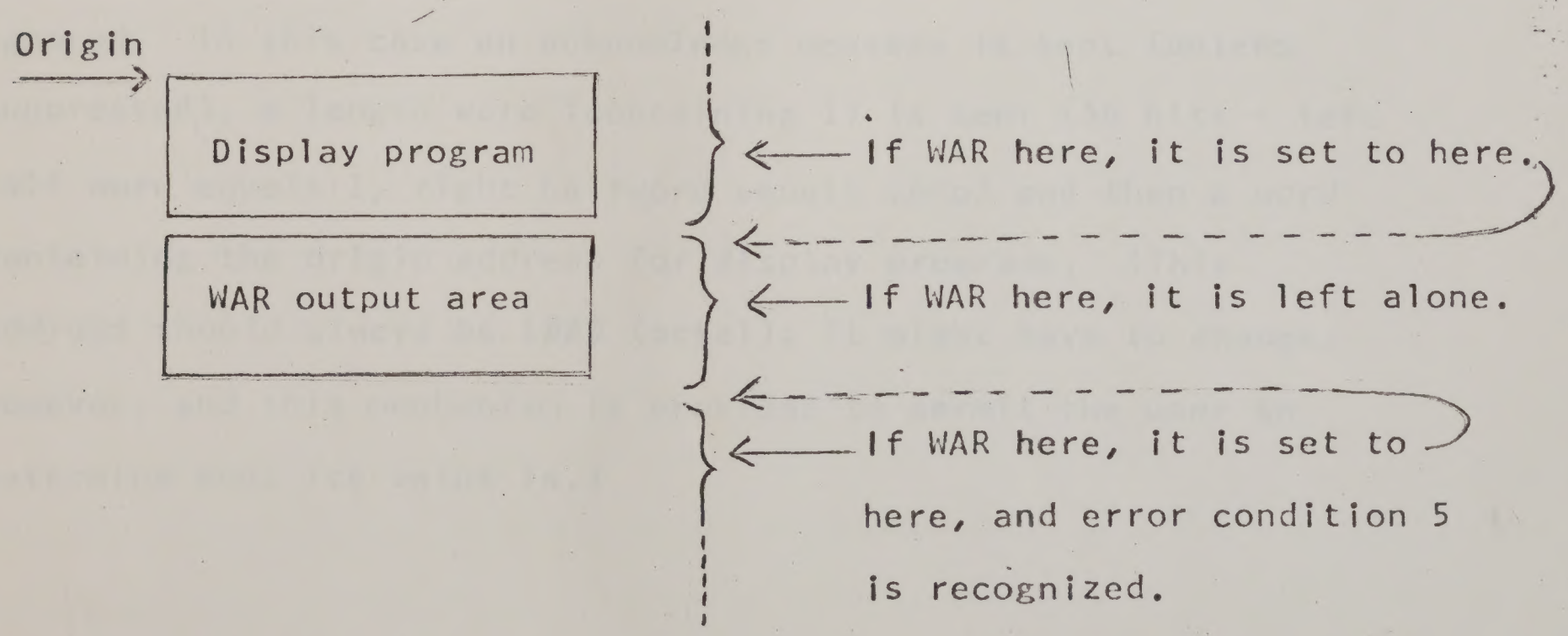
If the LDS-1 stops because too many output words are stored (i.e., if the WCR becomes positive or zero), error condition 5 is recognized. The number of output words made available to the user is as specified by A2 of the most recent SETUP request.

If the LDS-1 stops in any manner other than either those described above or by jumping to the word before the origin (the "finish" location), then error condition 7 is recognized. (A SETUP or an UPDATE is required before another EXECUTE is permitted.)

If the LDS-1 stops by jumping to the finish location, then the value of the WAR at the time determines the amount of output in the "WAR area" which the user may have access to.

If the WAR has been altered so that it contains an address smaller than its initial value, then the effective value of the WAR is its initial value. If the WAR contains a value greater than the address of the end of the area for output, then a WCR stop error is imitated (the effective value of the WAR is the maximum allowed by SETUP and error condition 5 is recognized). The situation in which no error is recognized will be called a "normal stop".

Summary for Normal Stop:


Origin
Display program
WAR output area
If WAR here, it is set to here.
If WAR here, it is left alone.
If WAR here, it is set to here, and error condition 5 is recognized.

If no error condition is recognized for an EXECUTE request, an acknowledge message is sent (unless suppressed; see MODE SET).

The effective value of the WAR is saved for later use in determining how much output the display program generated, but it is saved only for normal stops and WCR positive stops.

After normal stop, if Auto-TRANSMIT mode is set (see MODESET, below) then a TRANSMIT request is simulated (using the arguments specified in the MODESET request which initiated Auto-TRANSMIT mode). Otherwise the next request is begun.

3. The TRANSMIT Request.

Transmit takes one argument word and decodes the mode field, except in special cases. Normally A1 indicates the number of words of display program and/or output (in the WAR area) which are to be transmitted (possibly after data format conversion as indicated by the mode field). A2 indicates the starting address of the block of data to be (converted and) sent.

3A. Special Case: If A2 is zero, the mode and A1 are ignored. In this case an acknowledge message is sent (unless suppressed), a length word (containing 1) is sent (36 bits - left half word equals 1, right halfword equals zero) and then a word containing the origin address for display programs. (This address should always be 1ØØØ (octal); it might have to change, however, and this mechanism is provided to permit the user to determine what its value is.)

3B. The normal case is when A2 is non-zero.

If there has never been a SETUP request, error condition 1 is recognized. If the mode field is not zero or 1, or if A1 is negative, or A2 is less than the origin (but non-zero), or A1 + A2 is greater than the current "effective value of the WAR" (read on), then error condition 2 is recognized.

The "effective value of the WAR" always points to the word beyond the last word of output from the LDS-1. Before the first EXECUTE request, it points to the word after the end of the display program (indicating that zero words have been output). EXECUTE requests effect the value in various ways (depending on error conditions, etc.) as described previously. The effective WAR value is also changed to correctly reflect the effect of SETUP requests, which can change the size of the WAR area (type 1.B), and UPDATE requests, which can increase the size of the display program causing the whole WAR area to be shifted.

If the parameters are correct, an acknowledge message is sent (unless suppressed; see MODESET).

If A1, the number of words to be reformatted (possibly) and sent, is zero it is understood that the block of words to be processed starts at A2 and terminates with the last word before the effective WAR value. (Thus the number of words to be processed is "the effective WAR value minus A2".)

If the mode field of the first word of the request is zero then no reformatting will be done: an exact copy of the core image will be transmitted. If it is a one, then the block of words to be processed is <u>assumed</u> to contain the coordinates of

end points of a sequence of line segments. (The odd numbered words (starting counting with one, not zero!) are assumed to represent set-point data and the even-numbered words are assumed to contain draw-to data. (Note that this is consistent with LDS-1 "Scaled Coordinate to Memory" mode as long as no dots are displayed!!) The line segments are recoded in ARDS format, five seven-bit ASCII characters to a word, with the right most bit of each 36 bit word unspecified. The last word may contain null ASCII characters (all zero) to fill out the word.

The first word transmitted (after the acknowledge message, if any) has a left half of 3 and a right half of: Ø (for exact core image) or 1 (for ARDS format).

The next word transmitted has a left half equal to the number of words which follow and a right half of zero. For ARDS format, this is the number of words after reformatting (of course).

Unless zero words are to be transmitted, the next words are the data requested.

Note that the following sequence of requests could be used to simply convert line data (in the correct format) to ARDS format:

(The notation "A,,B" stands for a 36 bit word whose left half has the value "A" and whose right half has the value "B".)

```
        Establish network connection
        SETUP,,Ø                        ; start
        2n,,Ø                           ; data representing "n" line
                                        ;    segments follow
        +-------------------+           }
        |                   |           } 2n coordinate pairs
        +-------------------+           }
        TRANSMIT,,2                     ; request ARDS transmission
        Ø,,1ØØØ                         ; convert and transmit the
                                        ;    whole "program"
        FLUSH,,Ø                        ; signoff
```

4. The UPDATE request.

The UPDATE request allows a portion of the display program to be altered and also allows the end of the display to be extended (to include more data perhaps). The mode field of the first word of the request indicates the number of times the EXECUTE-action is to be performed if the UPDATE request is successful. The UPDATE request takes an argument word which is similar to that of TRANSMIT (i.e., A1 is a length and A2 is an address).

If no SETUP request has previously been executed error condition 1 is recognized.

If A1 is negative or if A2 is less than the origin of the display program, error condition 2 is recognized.

If A1 + A2 is greater than the end of the display program, then the length of the display program is increased to equal "A1 + A2 - origin". The WAR area remains the same length as before, and its contents are shifted to their new locations. The "effective WAR value" (see TRANSMIT) is increased by the amount of increase in display program length.

An acknowledge message is sent (unless suppressed) if no error condition is recognized (and if no problem with core size limitations arises). Also, the flag which is set by severe error conditions encountered during EXECUTE requests, is cleared by successful UPDATE requests.

The next "A1" words are read and stored in consecutive locations starting in location "A2". If A1 is zero, no words are read.

If the mode field of the first word of the UPDATE request is positive, it is taken as an iteration count for a multiple EXECUTE request. Otherwise, the next request is read from the Network.

5. The FLUSH request.

The Flush request takes no arguments and ignores the mode field of the first word of the request. No acknowledge message is sent. The network connection is broken and the process destroys itself.

6. The MODESET request.

The mode field of the first request word is decoded first. If it is zero or less, or if it is greater than (currently) 6, error condition 2 is recognized.

6A. If the mode field is 1, then acknowledge messages and ASCII error messages are not to be suppressed.

6B. If the mode field is 2, then acknowledge messages and ASCII error messages are to be suppressed.

6C. If the mode field is 3, exit from Auto TRANSMIT mode. (See 6D.)

6D. If the mode field is 4 or 5, whenever an EXECUTE has a normal stop, a TRANSMIT request is to be automatically performed.

The mode field for the transmit is to be 4 less than the mode field for this MODESET request. One (more) word is read from the network. It contains the arguments for the TRANSMIT. (That is, the word is saved and used when the TRANSMIT operation is performed. The values of the arguments are checked at that time, not during the MODESET request.

EXAMPLE

Suppose a display routine calculated and saved a different 3 dimensional transformation matrix each time it was executed. (It might be programmed to make an object appear to rotate.) The user of the routine typically would want to set up the matrices and then "let 'er rip" with ten or twenty (or more) executions. This could be done as follows:

```
SETUP,,0
length,,2000                    ; length of display pgm
                                ; large output area
   ___
   ___
   ___
program
   ___
   ___

MODESET,,5                      ; auto transmit in ARDS mode
0,,1000+length                  ; arguments for transmit
                                ;    meaning "current contents
                                ;    of output area"

EXQ,,10                         ; execute 10 times, sending
                                ;    the data each time

UPDATE,,20                      ; update, then execute 20 times
8,,address of matrix            ; arguments for update, to
                                ;    change a matrix
8 words  new matrix

etc.
```

The output from the Graphics Loader for this example would be as follows:

```
ACKNOWLEDGE    setup,setup          ; (See description of ACKN
                                    ;    for format in word)
ACKNOWLEDGE    modeset,modeset
ACKNOWLEDGE    transmit, exq
     output,,1                      ; output in ARDS mode
     leng1,,Ø
     {data

ACKNOWLEDGE    transmit,exq
     output,,1
     leng2,,Ø
     {data
          .
          .                         (In total, there are 10 sets of output.)
          .
ACKNOWLEDGE    update,update
ACKNOWLEDGE    transmit, update
     output,,1
     leng11,,Ø
     {data
          .
          .                         (In total, there are 20 sets of output.)
          .
```

ACKNOWLEDGE MESSAGES

These messages are each a single 36 bit word. They can be suppressed dynamically by the MODESET request. The left half of the 36 bit word identifies the fact that it is an acknowledge message by containing a 1. The right half is further divided into two 9-bit fields (quarters). The right 9-bit field (within the right half word) contains the opcode field of the last request sent by the user. Thus if an UPDATE request was recieved last, this field is a 4.

The left 9-bit field (within the right half word) contains the number of the action actually being performed. This differs from the opcode of the last request received only for Auto TRANSMIT mode and when a positive mode field is given in an UPDATE request (which is an EXECUTE count).

Possible pairs of these fields are given below. Note that Auto TRANSMIT mode sends only one acknowledge message which indicates both successful execution of the display program and correct parameters for the TRANSMIT operation.

Currently Executing:	Most Recent Request Received:	Comments:
SETUP	SETUP	; Sent before data words are read.
EXQ	EXQ	; Never sent in Auto-Transmit mode.
TRANSMIT	EXQ	; Only sent in Auto-Transmit mode.
TRANSMIT	TRANSMIT	
EXQ	UPDATE	; Never sent in Auto-Transmit mode.
TRANSMIT	UPDATE	; Only sent in Auto-Transmit mode.

```
UPDATE      UPDATE      ; Sent before data words are read.
MODESET     MODESET     ; When setting up Auto-Transmit mode,
                        ;    this is sent before the argument
                        ;    word for TRANSMIT is read.
```

Notes:

1) A MODESET request which suppresses acknowledge messages is never acknowledged. One which permits acknowledge messages is always acknowledged.

2) Requests which read data words (certain SETUP, UPDATE and MODESET requests) send acknowledge messages (unless suppressed) before reading the data words.

ERROR MESSAGES

These messages contain one or more words. If acknowledge messages are suppressed, then only one word is sent. (The assumption was made that brevity of response would also be desired for error messages if it were also desired for normal output.)

The first (or only) word contains a left half of 2. (Identifying this as an error message). The right half contains the number of the error condition recognized; these numbers are summarized below.

If the long form of the error message is being used, additional words are sent, the first containing a positive count of the remaining words in the left half and zero in the right half. Words following the count word contain five 7-bit ASCII characters per word (the last word may end in null characters)

which spell out a descriptive error message. These messages are summarized below:

Error Condition Number	Error Message
Ø	UNRECOGNIZED OPERATION CODE
1	PREVIOUS SETUP REQUIRED
2	INVALID MODE OR OPERAND FIELD
3	PREVIOUS EXECUTE FAILED
4	DISPLAY PROCESSOR OCCUPIED
5	OUTPUT AREA OVERRUN
6	DISPLAY ERROR: TWO SECOND TIME OUT
7	DISPLAY ERROR: IMPROPER DISPLAY STOP
8	DISPLAY ERROR: MEMORY PROTECTION VIOLATION

OUTPUT FROM TRANSMIT

This output consists of an identifying word (left half of 3, right half containing Ø for image mode and 1 for ARDS mode), a count word (left half indicating the number of words following this word, right half of zero) and zero or more data words in whatever data format the first word indicated.

Appendix I

The environment at the beginning of execution of the user's display program is as follows:

The LDS-1 is in program mode.

RAR/ Origin of user's program.

WAR/ First word of a block of core at least as long as requested via SETUP operations. This block immediately follows the last word of the display program.

PC/ Origin of user's program.

SP/ Address of first word beyond the end of a 200 (octal) word area of combined data sink and control register stack.

P1/ Unspecified.

P2/ Unspecified.

DSP/ First word of a 200 (octal) word area of combined data sink and control register stack.

UR/ Unspecified.

RCR/ Unspecified.

WCR/ Minus one plus the negative of the length of the "WAR area" as specified in the most recent SETUP request.

DIR/	Stop on WCR positive.
	Do not stop on Hit.
	Matrix Multiplier inactive.
	"Overlap" permitted.
	Two dimensional mode.
	Do not "Do Twice".
RSR/	Unspecified.
SR/	Ø, except for bits not under program control
	like the stylus "Z" coordinate bits.
NEXT/	Unspecified.
SAVELB/	-3777,,-3777
SAVERT/	3777,,3777
VIEWLB/	-3777,,-3777
VIEWRT/	3777,,3777
WINDLB/	-3777,,-3777
WINDRT/	3777,,3777
INSTLB/	-3777,,-3777
INSTRT/	3777,,3777
NAME/	Unspecified.
CDIR/	Scaled Coordinates to Memory
	Curve mode inactive.
	Minimum effort inactive.
	Solid, not dashed, lines.
	Self mode inactive.

HITANG/ Unspecified.

SELINT/ Unspecified.

Matrix Multiplier/ Unspecified.

Origin of program -1/ (Should be jumped to when the display program is finished.)